

ELETTRONICA GENERALE, FONDAMENTI DI ELETTRONICA DIGITALE
Appello d'esame del 5/9/2016

Ogni risposta corretta +2 punti, ogni risposta sbagliata -0,5 punti, ogni risposta in bianco 0 punti
Minimo 6 punti sulle domande, minimo 2 punti sui problemi (20 minuti)

Cognome _____ Nome _____ Mat. _____ Data __/__/__

- 1) Itn rappresenta...
 - a la corrente assorbita da uno stadio Schmidt trigger durante la commutazione verso "0"
 - b la corrente assorbita dall'uscita in una transizione verso "0"
 - c la corrente assorbita dall'alimentazione quando le uscite sono a "0"

- 2) I dispositivi three-state...
 - a differiscono dai dispositivi normali quando l'uscita è a "1"
 - b hanno un'uscita ben posta anche se l'ingresso è in zona di incertezza ($V_{il} < V_{in} < V_{ih}$)
 - c differiscono dai dispositivi normali perché aggiungono in uscita un terzo stato "Z"

- 3) La tecnologia ... modifica lo stadio di uscita rispetto alle tecnologie bipolari precedenti.
 - a RTL
 - b DTL
 - c TTL

- 4) Un comparatore a 4 bit
 - a ha solo una linea di uscita che si attiva in caso gli ingressi siano uguali
 - b è un dispositivo sincrono con tutti i flip-flop con il clock collegato insieme
 - c ha opportuni ingressi e uscite per poter essere connesso in cascata e formare operatori a byte, a word, ecc.

- 5) Un flip-flop differisce da un latch perché
 - a abilita l'ingresso "in" sul fronte del segnale "ck/en" invece che sul livello
 - b è attivo sul fronte di salita del segnale "ck/en" invece che sul fronte di discesa
 - c non soffre del problema della metastabilità

- 6) Il convertitore A/D sigma/delta ...
 - a è il più veloce dei convertitori
 - b filtra passa-basso il segnale e passa-alto il rumore, migliorando l'SNR
 - c è in disuso perché soppiantato dal convertitore ad approssimazioni successive

- 7) Un dispositivo PLE...
 - a implementa solo forme canoniche
 - b è il dispositivo logico programmabile più recente
 - c permette l'implementazione di contatori sincroni e asincroni

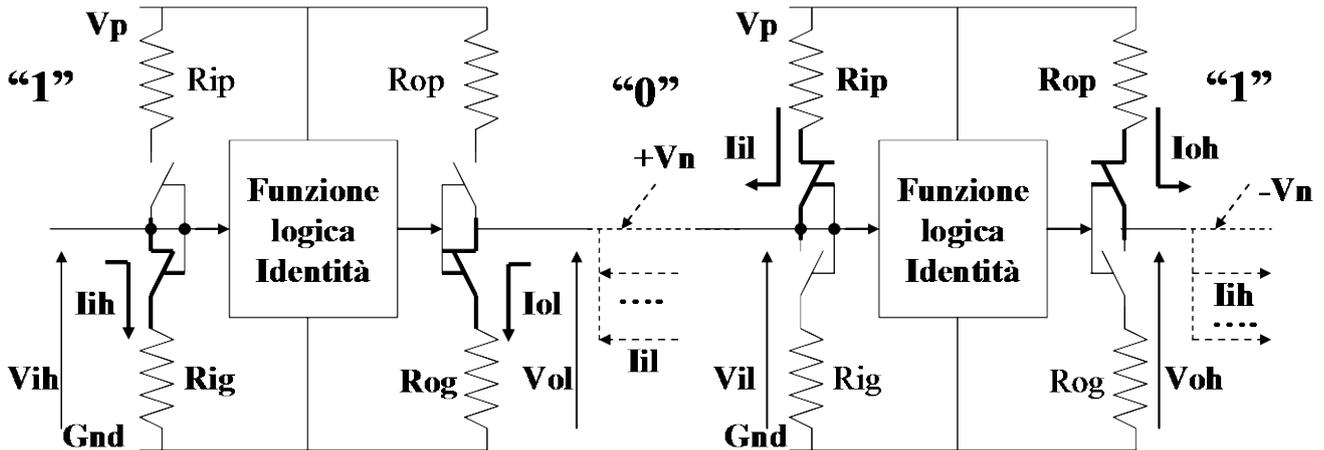
ELETTRONICA GENERALE, FONDAMENTI DI ELETTRONICA DIGITALE
Appello d'esame del 5/9/2016 (1h 40 minuti)

Problema

Dato il modello in figura si definisca l'immunità al rumore V_n e il FAN-OUT statico N_{max} (1 punto)

Si definisca la zona d'incertezza (1 punto)

Dimostrare perché si deve avere $R_{og} \ll R_{ip}$ (e analogamente $R_{op} \ll R_{ig}$) (1 punto)



Problema

1) Spiegare il funzionamento della porta logica riportata in figura 1 a sinistra scrivendo l'espressione di F in forma SOP. (1 punto)

2) Illustrare le implementazioni in fig. 2 e in fig. 3 in termini di funzione logica realizzata. (1 punto)

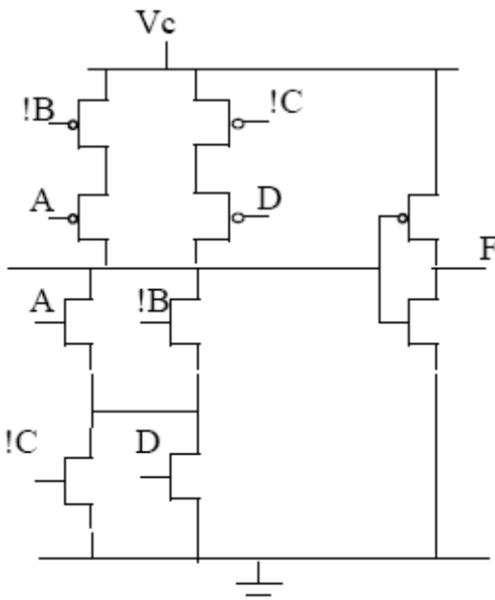


Figura 1

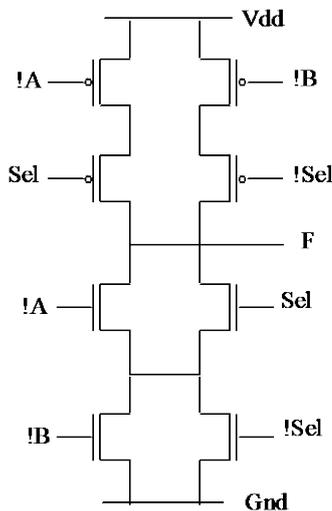


Figura 2

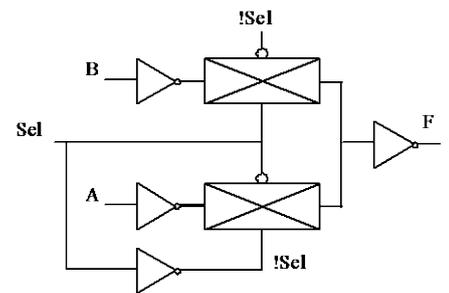


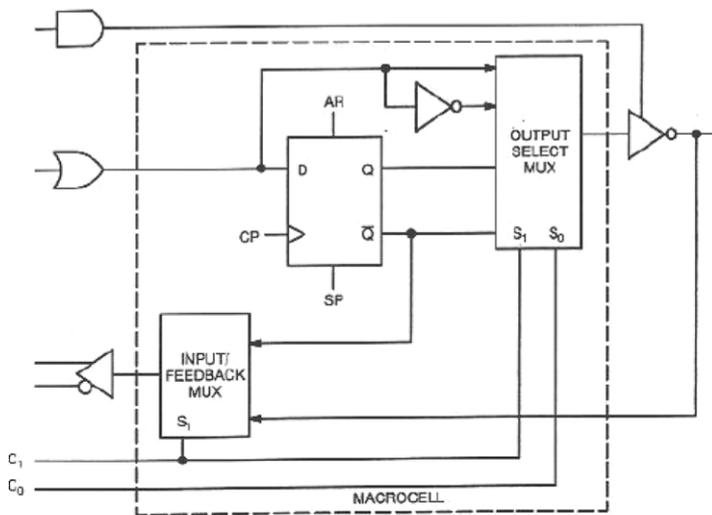
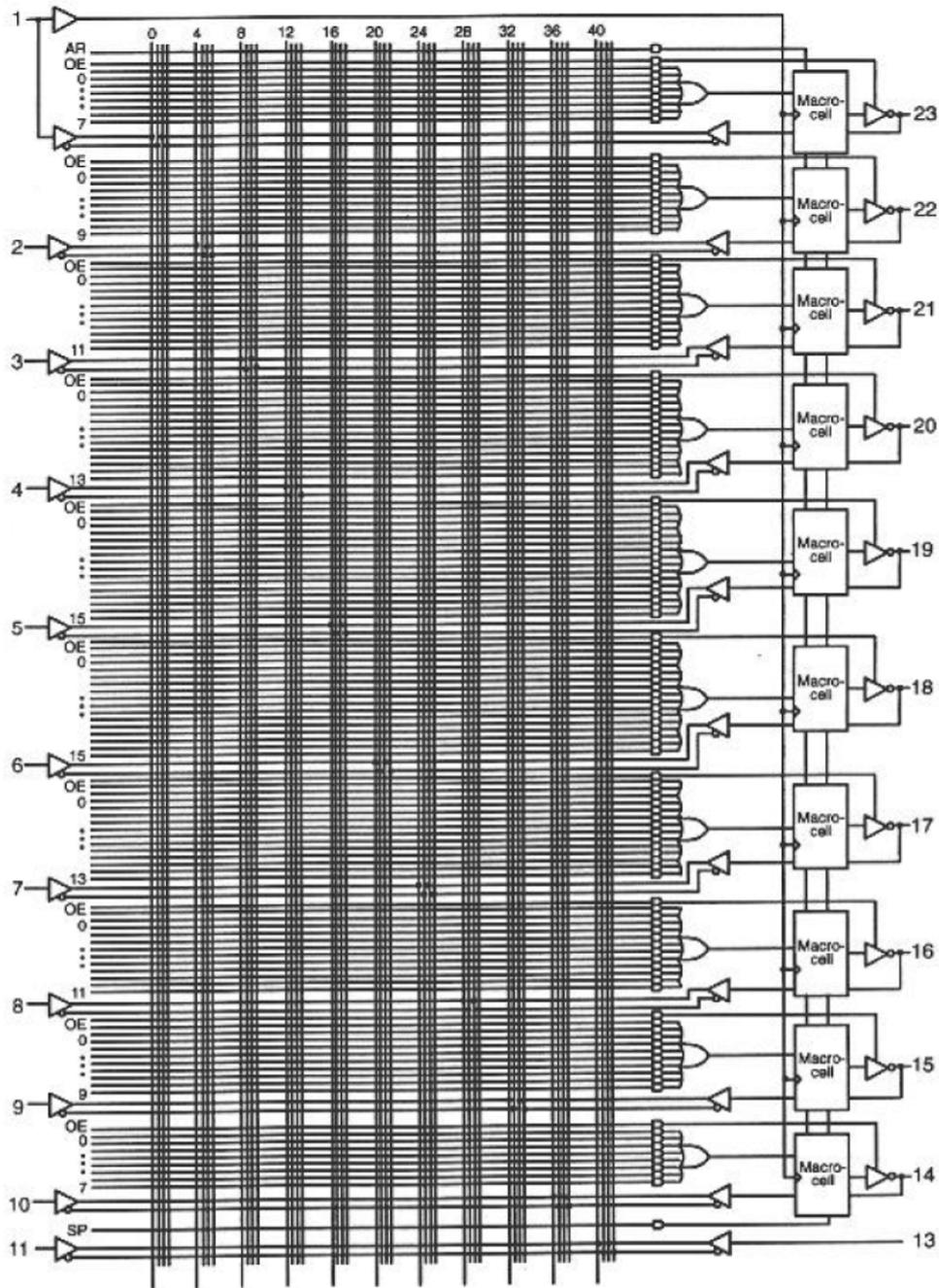
Figura 3

Problema

1) Si motivi la necessità di usare il tipo `std_logic` piuttosto che il `bit` in VHDL per rappresentare una porta di tipo three-state (1 punto)

2) Si definisca il priority encoder a 7 ingressi attivi alti e tre uscite attive alte. Si sviluppi il programma in linguaggio booleano per una Gal22V10 (1 punto)

3) ...e il programma VHDL (1 punto)



SOLUZIONI

ELETTRONICA GENERALE, FONDAMENTI DI ELETTRONICA DIGITALE

Appello d'esame del 5/9/2016

Ogni risposta corretta +2 punti, ogni risposta sbagliata -0,5 punti, ogni risposta in bianco 0 punti

Minimo 6 punti sulle domande, minimo 2 punti sui problemi (20 minuti)

Cognome _____ Nome _____ Mat. _____ Data __/__/__

1) Itn rappresenta...

- a la corrente assorbita da uno stadio Schmidt trigger durante la commutazione verso "0"
- b la corrente assorbita dall'uscita in una transizione verso "0"
- c la corrente assorbita dall'alimentazione quando le uscite sono a "0"

2) I dispositivi three-state...

- a differiscono dai dispositivi normali quando l'uscita è a "1"
- b hanno un'uscita ben posta anche se l'ingresso è in zona di incertezza ($V_{il} < V_{in} < V_{ih}$)
- c differiscono dai dispositivi normali perché aggiungono in uscita un terzo stato "Z"

3) La tecnologia ... modifica lo stadio di uscita rispetto alle tecnologie bipolari precedenti.

- a RTL
- b DTL
- c TTL

4) Un comparatore a 4 bit

- a ha solo una linea di uscita che si attiva in caso gli ingressi siano uguali
- b è un dispositivo sincrono con tutti i flip-flop con il clock collegato insieme
- c ha opportuni ingressi e uscite per poter essere connesso in cascata e formare operatori a byte, a word, ecc.

5) Un flip-flop differisce da un latch perché

- a abilita l'ingresso "in" sul fronte del segnale "ck/en" invece che sul livello
- b è attivo sul fronte di salita del segnale "ck/en" invece che sul fronte di discesa
- c non soffre del problema della metastabilità

6) Il convertitore A/D sigma/delta ...

- a è il più veloce dei convertitori
- b filtra passa-basso il segnale e passa-alto il rumore, migliorando l'SNR
- c è in disuso perché soppiantato dal convertitore ad approssimazioni successive

7) Un dispositivo PLE...

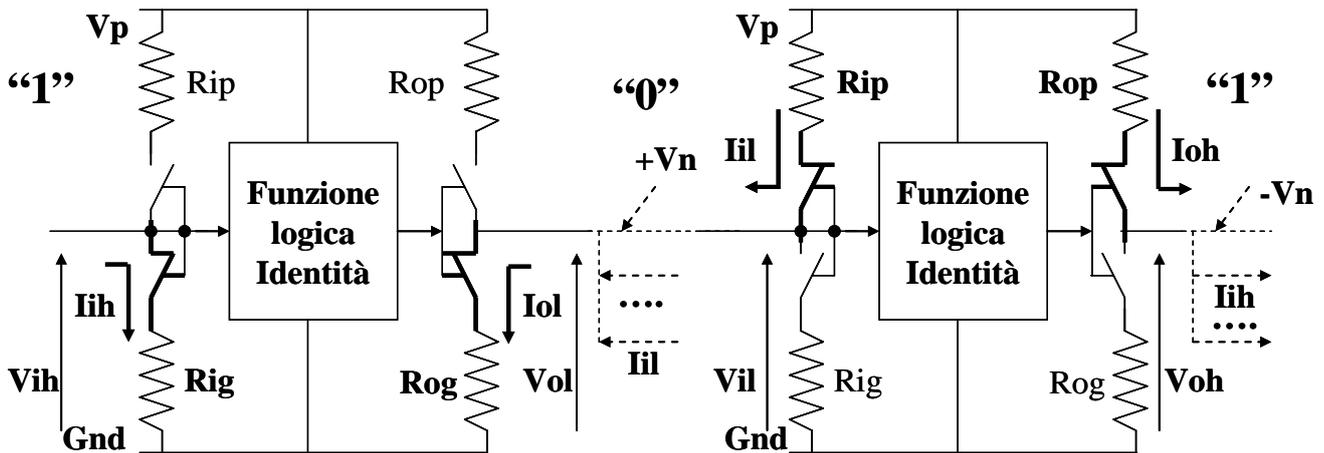
- a implementa solo forme canoniche
- b è il dispositivo logico programmabile più recente
- c permette l'implementazione di contatori sincroni e asincroni

ELETTRONICA GENERALE, FONDAMENTI DI ELETTRONICA DIGITALE
Appello d'esame del 5/9/2016

Soluzioni

Problema

Dato il modello in figura si definisca l'immunità al rumore V_n e il FAN-OUT statico N_{max} (1 punto)
 Si definisca la zona d'incertezza (1 punto)
 Dimostrare perché si deve avere $R_{og} \ll R_{ip}$ (e analogamente $R_{op} \ll R_{ig}$) (1 punto)



Soluzione

L'immunità al rumore V_n è il valore del massimo segnale di tensione (rumore) che può essere sommato o sottratto all'uscita di un dispositivo garantendo il corretto riconoscimento del segnale logico in ingresso

$$V_n = \min (V_{il} - V_{ol} , V_{oh} - V_{ih}) \quad \text{Normalmente } V_n > 0,1 \text{ V}$$

Il FAN-OUT statico N_{max} corrisponde al massimo numero di carichi che possono essere connessi ad un dispositivo garantendo il corretto riconoscimento del segnale logico in ingresso

$$N_{max} = \min (I_{oh} / I_{ih} , I_{ol} / I_{il}) \quad \text{Normalmente } N_{max} > 10$$

La zona d'incertezza corrisponde a quel range di valori di tensione compresi tra V_{il} e V_{ih} che, se applicati all'ingresso di un dispositivo, non producono un valore predicibile in uscita.

In uscita alla prima porta si ha

$$V_{ol} = R_{og} * I_{ol} = R_{og} * \sum I_{il}$$

Ma $I_{il} = (V_p - V_{ol}) / R_{ip}$ e quindi si ha

$$V_{ol} = R_{og} * \sum (V_p - V_{ol}) / R_{ip}$$

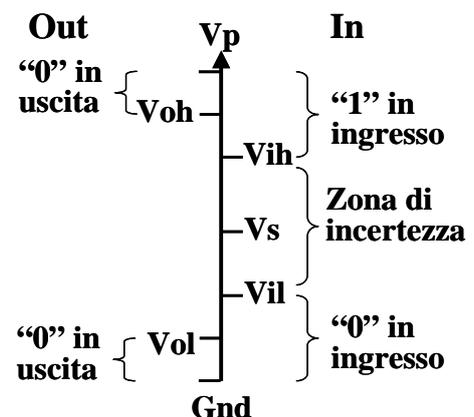
Se si suppongono N carichi allora

$$V_{ol} = N * (V_p - V_{ol}) * R_{og} / R_{ip}$$

$$R_{og} / R_{ip} = (1/N) * V_{ol} / (V_p - V_{ol})$$

Ma $V_{ol} / (V_p - V_{ol}) < 1$ in quanto V_{ol} è più vicina a gnd che a V_p ($V_{ol} \sim 0V$); dato che un dispositivo deve poter pilotare tanti dispositivi ($N > 10$) allora $R_{og} / R_{ip} \ll 1$ e cioè $R_{og} \ll R_{ip}$

Infatti i dispositivi digitali hanno alta resistenza di ingresso e bassa resistenza di uscita.



Problema

- 1) Spiegare il funzionamento della porta logica riportata in figura 1 a sinistra scrivendo l'espressione di F in forma SOP. (1 punto)
- 2) Illustrare le implementazioni in fig. 2 e in fig. 3 in termini di funzione logica realizzata. (1 punto)

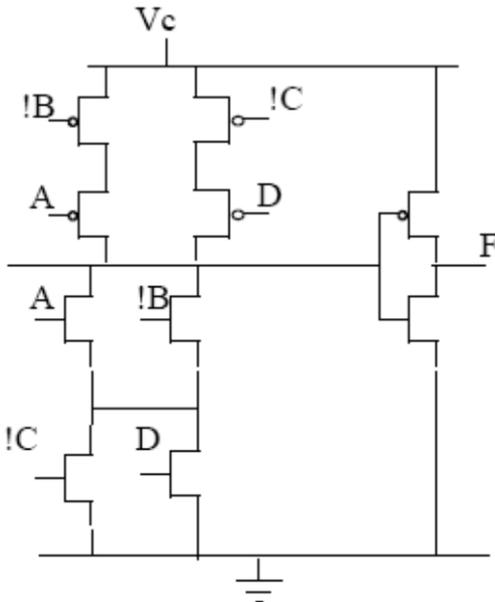


Figura 1

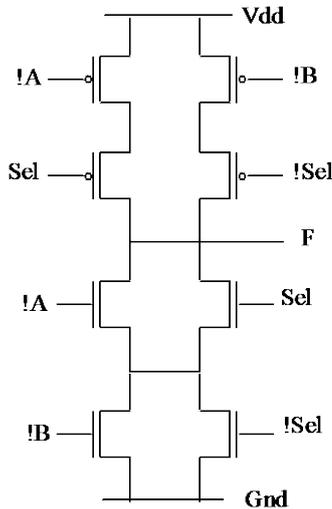


Figura 2

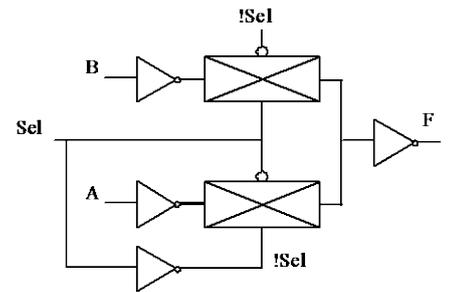


Figura 3

Soluzione

1) La porta logica mostrata realizza una funzione F in logica FCMOS. Considerando che è stato inserita una porta NOT prima dell'uscita F, quindi le equazioni finali sono negate, si può scrivere:

$$!F = !A \& B + C \& !D \quad F = (A + !B) \& (!C + D) = A \& !C + A \& D + !B \& !C + !B \& D$$

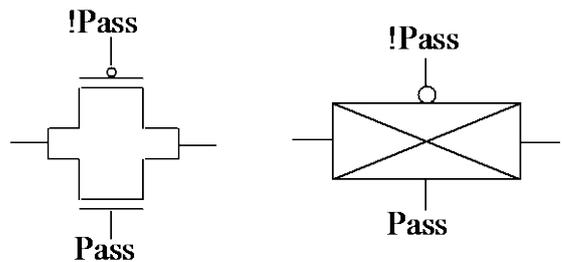
2) Il circuito in figura 2 implementa la funzione multiplexer in forma non "buffered" e in particolare si ha:

$$F = A \& !Sel + B \& Sel$$

Il circuito in Figura 3 implementa la stessa funzione ma è "buffered" e utilizza elementi switch. L'uso di elementi switch richiede necessariamente l'architettura "buffered" ma in taluni casi può risultare vantaggiosa in termini di occupazione di spazio silicio. Nel caso in esame si avrebbe:

Figura 2: 7 PMOS + 7 NMOS (conteggiando anche le porte NOT che servono per ottenere !A, !B e !Sel).

Figura 3: 6 PMOS + 6 NOMS



Problema

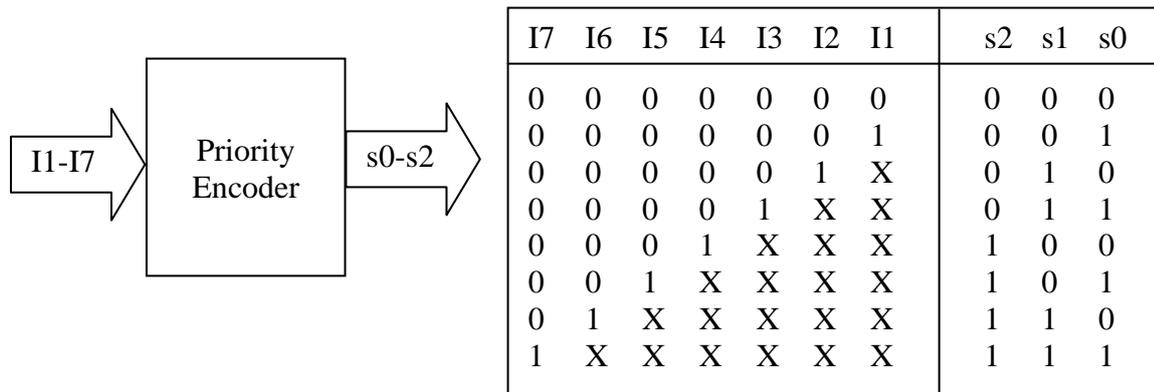
- 1) Si motivi la necessità di usare il tipo **std_logic** piuttosto che il **bit** in VHDL per rappresentare una porta di tipo three-state (1 punto)
- 2) Si definisca il priority encoder a 7 ingressi attivi alti e tre uscite attive alte. Si sviluppi il programma in linguaggio booleano per una Gal22V10 (1 punto)
- 3) ...e il programma VHDL (1 punto)

Soluzione

1) Un dispositivo a tre stati può fornire in uscita un valore di tensione alto o basso a seconda che debba rappresentare uno '0' o un '1' logico e, se disabilitato, un altro valore detto Z o stato ad alta impedenza (che corrisponde in pratica a disconnettere il dispositivo dalla linea). Un segnale di tipo bit può assumere solamente i valori logici '0' e '1' e non può quindi essere usato per rappresentare quest'ultima condizione, a differenza dello std_logic che prevede 9 livelli tra i quali anche lo stato Z.

2) La funzione "priority encoder" è un dispositivo che traduce una codifica espansa in una compatta. A seconda della linea di ingresso attiva, in uscita viene generata una codifica univoca, generalmente la codifica binaria del numero dell'ingresso attivo. Nel caso nessun ingresso risulti attivo, generalmente le uscite risultano tutte "basse". Se più di un ingresso per volta è attivo, in uscita viene generata la codifica dell'ingresso attivo con priorità maggiore, secondo una ben determinata scala.

Lo schema a blocchi e la tabella della verità del priority encoder richiesto (I7 con massima priorità) sono i seguenti.



Dalla tabella della verità si ricavano le seguenti equazioni booleane:

$$s2 = I7 + !I7 \& I6 + !I7 \& !I6 \& I5 + !I7 \& !I6 \& !I5 \& I4$$

$$s1 = I7 + !I7 \& I6 + !I7 \& !I6 \& !I5 \& !I4 \& I3 + !I7 \& !I6 \& !I5 \& !I4 \& !I3 \& I2$$

$$s0 = I7 + !I7 \& !I6 \& I5 + !I7 \& !I6 \& !I5 \& !I4 \& I3 + !I7 \& !I6 \& !I5 \& !I4 \& !I3 \& !I2 \& I1$$

Il programma per il dispositivo GAL22V10 risulta quindi:

GAL22V10

```
nc    I1    I2    I3    I4    I5    I6    I7    nc    nc    nc    GND
nc    nc    nc    nc    s2    s1    s0    nc    nc    nc    nc    VCC
```

EQUATIONS

$$s2 = I7 + !I7 \& I6 + !I7 \& !I6 \& I5 + !I7 \& !I6 \& !I5 \& I4$$

$$s1 = I7 + !I7 \& I6 + !I7 \& !I6 \& !I5 \& !I4 \& I3 + !I7 \& !I6 \& !I5 \& !I4 \& !I3 \& I2$$

$$s0 = I7 + !I7 \& !I6 \& I5 + !I7 \& !I6 \& !I5 \& !I4 \& I3 + !I7 \& !I6 \& !I5 \& !I4 \& !I3 \& !I2 \& I1$$

$$s2.OE = Vcc$$

```
s1.OE = Vcc  
s0.OE = Vcc
```

3) Soluzione in VHDL

```
LIBRARY ieee;  
USE ieee.std_logic_1164.ALL;  
  
ENTITY prienc IS  
    PORT (I7, I6, I5, I4, I3, I2, I1: IN std_logic;  
          S: OUT std_logic_vector(2 DOWNTO 0));  
END prienc;  
  
ARCHITECTURE archprienc OF prienc IS  
BEGIN  
  
    enc: PROCESS (I7,I6,I5,I4,I3,I2,I1)  
    BEGIN  
        IF I7 = '1' THEN S <= "111";  
        ELSIF I6 = '1' THEN S <= "110";  
        ELSIF I5 = '1' THEN S <= "101";  
        ELSIF I4 = '1' THEN S <= "100";  
        ELSIF I3 = '1' THEN S <= "011";  
        ELSIF I2 = '1' THEN S <= "010";  
        ELSIF I1 = '1' THEN S <= "001";  
        ELSE S <= "000";  
        END IF;  
    END PROCESS enc;  
  
END archprienc;
```